

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('/content/Crop_recommendation.csv')
df.head()
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

```
df.info()
```

```
if df['N'].all()>90:
    print(df['N'])
```

```
df.isnull().sum()
```

```

N          0
P          0
K          0
temperature 0
humidity     0
ph           0
rainfall     0
label        0
dtype: int64
```

```
x=df.drop('label',axis=1)
y=df['label']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test, y_train,y_test = train_test_split(x,y,stratify=y,random_state=1)
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
from sklearn.metrics import accuracy_score
logistic_acc = accuracy_score(y_test,y_pred)
print("Accuracy of logistic regression is "+ str(logistic_acc))
```

```
Accuracy of logistic regression is 0.9618181818181818
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
https://scikit-learn.org/stable/modules/preprocessing.html
```

```
Please also refer to the documentation for alternative solver options:
```

```
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
```

```
n_iter_i = _check_optimize_result(
```

```
from sklearn.tree import DecisionTreeClassifier
model_2 = DecisionTreeClassifier(criterion='entropy',max_depth = 6,random_state=2)
model_2.fit(x_train,y_train)
y_pred_2 = model_2.predict(x_test)
decision_acc = accuracy_score(y_test,y_pred_2)
print("Accuracy of decision tree is "+str(decision_acc))
```

```
Accuracy of decision tree is 0.9781818181818182
```

```

from sklearn.naive_bayes import GaussianNB
model_3 = GaussianNB()
model_3.fit(x_train,y_train)
y_pred_3 = model_3.predict(x_test)
naive_bayes_acc = accuracy_score(y_test,y_pred_3)

from sklearn.ensemble import RandomForestClassifier
model_4 = RandomForestClassifier(n_estimators = 25,random_state = 2)
model_4.fit(x_train.values,y_train.values)
y_pred_4 = model_3.predict(x_test)
random_fore_acc = accuracy_score(y_test,y_pred_4)
print("Accuraacy of Random Forest is "+str(random_fore_acc))

```

Accuraacy of Random Forest is 0.9945454545454545

```
import joblib
```

```
file_name = 'crop_app'
```

```
joblib.dump(model_4,'crop_app')
```

```
['crop_app']
```

```
app = joblib.load('crop_app')
```

```
arr = [[90,42,43,20.879744,82.002744,6.502985,202.935536]]
acc = app.predict(arr)
```

```
acc
```

```
array(['rice'], dtype=object)
```

```
import pickle
```

```
import pickle
```

```
Pkl_Filename = "Pickle_RL_Model.pkl"
```

```
with open(Pkl_Filename, 'wb') as file:
    pickle.dump(model_4, file)
```

```
with open(Pkl_Filename, 'rb') as file:
    Pickled_Model = pickle.load(file)
```

```
Pickled_Model
```

```

RandomForestClassifier
RandomForestClassifier(n_estimators=25, random_state=2)

```